Understanding Elasticity of Cloud Services Compositions

Thar Baker
School of Computing, Maths and Digital Technology
Manchester Metropolitan University
Manchester, UK
t.baker@mmu.ac.uk

Azzelarabe Taleb-Bendiab
School of Computer and Security Science
Edith Cowan University, Perth, AU
a.taleb-bendiab@ecu.edu.au

Martin Randles
School of Computing and Maths Sciences
Liverpool John Moores University
Liverpool, UK
m.j.randles@ljmu.ac.uk

Aseel Hussien
School of Built Environment
Liverpool John Moores University
Liverpool, UK
a.hussien@ljmu.ac.uk

Abstract- This paper presents a new Intention Description Layer (IDL), which supports the Provision, Assurance, and Accounting (PAA) modelling approach, to enable the weaving of a new service/component to the cloud application at runtime to achieve the elasticity nature of the cloud. The new layer aims at bridging the technical details required to be understood well before injecting any service into a running application, by the user. This will be in terms of how, why and what is done by representing the business processes managed by business rules.

I. INTRODUCTION AND PROBLEM STATEMENT

The key beneficial of the cloud computing is the user/client since the cloud platforms and infrastructure rent is much cheaper than buying/investing full hardware to achieve a certain benefit. For this, it should be easy to use from the end user point of view. These kinds of platforms are becoming widely used in dynamic economic environments, where a company’s survival depends on its applications’ status and the applications admin’s ability to focus on core business needs and adapt the application quickly at work time by adding, removing and modifying any of the web services or components based on the BPEL and WSDL [1].

However, the Web Service Description Language (WSDL) [2, 3] describes the technical details and interoperability issues such as (services partners, partner link types and port types), to show how a web service can be accessed, invoked and linked to other services over the web to expand systems functionalities. In addition, state of the art business process automation systems are based on Business Process Execution Language (BPEL) [4] as services composition language, which relies on WSDL for describing and automating business behaviour. Moreover, BPEL and WSDL were both proposed to be used by computers, therefore it defines only how the services could be linked technically [5]. However, as shown in [6], the interoperability of cloud computing services requires a lot more services’ details and description than what is listed in WSDL and used by BPEL. To solve this problem, we proposed a new Intention Description Layer (IDL) to the cloud stack that can be used via Provision, Assurance, and Accounting (PAA) modelling approach, to weave a new components/services to the cloud application at runtime.

II. A PROPOSED SOLUTION

This paper presents an approach supported by two elements: (i) a new IDL and its model, which is easily accessible and used by the cloud user. The new layer contents can be easily adapted at runtime to accommodate any runtime adaptations of a given application. (ii) the PAA framework as a service that provides a cloud-based interpreter to execute the IDL model, which generates a cloud application and/or enables the adaptation of a managed cloud application. While the PAA framework has been widely explained in [7], this paper focuses on detailing IDL only.

A. IDL Description

IDL is an xml-based cloud application description layer. It uses three main approaches to model and automate business behaviour on cloud applications. First approach focuses on flows of activities that generate values, which lead to the output/results of the application. This flow model will basically be used to describe “what” is happening to get this final results; second approach uses rules to describe desired business behaviour. This so called Business Rules approach uses rules to describe “why” something has to happen according to the rules and the requirements, and uses technologies to automate decision logic. Third one deals with the technical specification of the services and components that should be used to accomplish certain behaviour.

B. Formalising IDL Meta-model

The IDL consists mainly of a set of processes that describe the business behaviours. Some of these processes are composed of sub processes which both include a task or set of tasks that should be executed to achieve the desired behaviour of that process. The process model information is attached to the IDL to provide full information about the current IDL, for example, the IDL creator, IDL name, IDL owner, and the guide key. There are different kinds of requirements, listing 1, that are defined in the IDL to give a clear view to the user and to satisfy the three different approaches explained above, (1) The Provision requirements show the specific user requirements and the individual services and components the user wants to use in the application at runtime. (2) The Assurance requirements describe properties that are fundamental for the characterization of a service. (3) The Accounting requirements represent the statistical side of the application and supply the dashboard with the necessary...
information to allow better managing of the application at runtime.

Listing 1: PAA requirements

```
define ValidPreShippingOrderPAA as PAA
{
  provisioning
  {
    order as PetOrderDetails
  }
  assurance
  {
    order.ShippingAddress is VALID
    order.SelectedPet is INSTOCK
    order.ShippingDate is NULL
  }
  Accounting
  {
    /*error handling or logging defined here*/
  }
```

III. CASE STUDY: PETSHOP IDL

The process model of the standard Microsoft PetShop blueprint does not support EC2 and cannot be adapted at runtime. The whole system will need to be taken offline for the adaptation, and re-upload it after the adaption. Since the system admin should undertake this, the user will not be able to add any new function or service to the original process model on the cloud. Hence, PetShop is re-designed using the PetShop cloud IDL. The user will be dealing with IDL and the movement from PetShop web page to the next one is managed and controlled by IDL flow model.

IV. EVALUATION: SYSTEM SCALABILITY

A test was conducted to ascertain the scalability of cloud application PetShop during its operation in comparison to the original Microsoft PetShop. By introducing a new behaviour to new PetShop via IDL, the performance impact of the further interpretation of the new behaviour could be contrasted against that of the performance impact of the introduction of the same behaviour in the Microsoft PetShop model.

Figure 1 shows two systems executed to produce the same behaviours over a time-scale of cycles of a process execution. After 50,000 executions, a new task requirement is introduced in the IDL model, and a task description updated accordingly. As new semantic linking needs to take place, performance is reduced, such that the time to complete the process increases. It should be noted, however, that as only one new requirement is needed to be linked, the performance impact is less than that of the original initiation at cycle 0, where many new requirements are introduced at the same time. After the reconfiguration, linking and execution, performance returns to a new standard, slightly slower than the original behaviour from 10,000 to 50,000. This is due to the added time needed to execute the action by PAA.

On the other hand, there is a slight increase in execution time of the original PetShop (given in green) from 50,000, due to the new behaviour introduced to the PetShop code to be executed. It can be noted that after the execution of the new behaviour, however, the execution performance of PAA PetShop compared to Microsoft PetShop is largely similar, thus scalability issues can be said to be introduced.

```
Figure 1: Scalability testing of PAA PetShop vs. Microsoft PetShop
```

V. CONCLUSION AND FUTURE WORK

This paper has argued for the design and use of an adaptable intention description layer for cloud-based–applications to achieve the cloud elasticity characteristics through the using of the PAA approach. This means it will be straightforward, quick and reliable to amend the cloud-based-applications at run time without the need for the recoding and republishing of the application.

The perceived value of using IDL via PAA has driven the re-development of PetShop to be a Cloud-Based-Application through this approach.

In future work, the IDL description is to be rigorously tested and evaluated to ensure the design of IDL and PAA tools is correctly envisaged. But there is still a large body of outstanding research issues, which have to be resolved before the full benefits of exploitation can be realised.

REFERENCES